

CLAIMS

I claim:

1. A method of encoding a data file in a sequence of video frames, said file having a name and comprising a sequence of bits contained within a pre-determined number of bytes, said method comprising the steps of:

creating a first sequence of data packets each comprising a header field, a user data field comprising a first pre-determined number of bits, and a checksum field;

partitioning said sequence of bits into a first sequence of blocks each comprising a number of bits less than or equal to said first pre-determined number of bits;

associating each respective block in said first sequence of blocks with a respective data packet in said first sequence of data packets;

copying each said block into the user data field of said respective data packet;

creating a metadata packet comprising a header field, a user data field comprising said first pre-determined number of bits, and a checksum field;

encoding information in the user data field of said metadata packet comprising said name and said pre-determined number of bytes;

creating a second sequence of data packets comprising said first sequence and the metadata packet;

associating each data packet in said second sequence with a respective video frame in said sequence of video frames;

partitioning each data packet in said second sequence into a second sequence of blocks each comprising a second pre-determined number of bits;

creating a sequence of binary numbers by examining said second sequence of blocks and interpreting said second pre-determined number of bits as a binary number;

allocating respective regions of said respective video frame to each number in said sequence of binary numbers;

assigning video component levels associated with each said binary number to each said region to generate said respective video frame; and

outputting each video frame in said sequence of video frames whereby the data file is represented as a sequence of video frames.

2. A method according to claim 1, wherein the header in said data packet contains a number associated with said data file and the header in said metadata packet contains a number associated with said data file.
3. A method according to claim 1, wherein the header in said data packet contains a number associated with a particular block in said first sequence of blocks.
4. A method according to claim 1, wherein the video frame is output to a graphics file containing one image or a plurality of images.
5. A method according to claim 1, wherein the video frame is output to a storage device.
6. A method according to claim 5, wherein the storage device is videotape.
7. A method according to claim 1, wherein a look up table is used to assign said video component levels.
8. A method according to claim 1, wherein pre-determined regions of said respective video frame are assigned pre-determined video component levels.

9. A method according to claim 1, wherein an integral number of data packets is associated with a video frame.
10. A method of decoding a data file encoded in a sequence of video frames, said file having a name and comprising a first sequence of bits contained within a pre-determined number of bytes, said method comprising the steps of:
 - inputting a sequence of video frames each comprised of a pre-determined sequence of pre-determined regions comprised of pixels;
 - assigning a set of video component levels to each region in said pre-determined sequence by sampling the video component levels of each of the said pixels and combining the sampled levels;
 - creating a sequence of binary numbers each comprised of a first pre-determined number of bits by associating a respective binary number with each said region;
 - creating a sequence of bits by combining the bits comprising each said respective binary number;
 - partitioning said sequence of bits into a first sequence of data packets each comprising a header field, a user data field, and a checksum field;
 - validating each packet in said first sequence of data;
 - creating a second sequence of data packets comprised of valid data packets contained in said first sequence of data packets;
 - dividing said second sequence into a metadata packet and a plurality of normal data packets;
 - creating a first data file;

copying bits from said normal data packets to said first data file;

decoding the bits in the user data field of said metadata packet to determine a file name and a file size;

creating a second data file;

associating said file name and said second data file;

copying a number of bytes equal to said file size from the first data file to the second data file whereby said data file encoded in a sequence of video frames is decoded.

11. A method according to claim 10, wherein performing a checksum calculation validates packets.
12. A method according to claim 10, wherein a packet is validated by sampling pre-determined video component levels at pre-determined regions of a video frame that encodes the packet.
13. A method according to claim 10, wherein the position of pre-determined regions within a video frame is determined by a method comprised of sampling pre-determined video component levels.
14. A method according to claim 10, wherein a set of video component levels is determined by a method comprised of sampling video component levels at pre-determined regions of a video frame.
15. A method according to claim 10, wherein said file name and file size is determined using bits in the header fields of said first sequence of data packets.
16. A method according to claim 10, wherein the file size is indeterminate.

17. A method of broadcasting a data file to a plurality of receivers comprising the steps of:

Encoding said data file in a sequence of video frames;

Creating a television program comprising said sequence of video frames;

Delivering said television program to a television station;

Causing said television station to broadcast said television program.

18. A method according to claim 17, wherein said television program comprises images created by superimposing each video frame in said sequence of video frames on a respective video frame comprised of an ordinary video image.